Bayesian Verification and Validation for Adaptive Systems

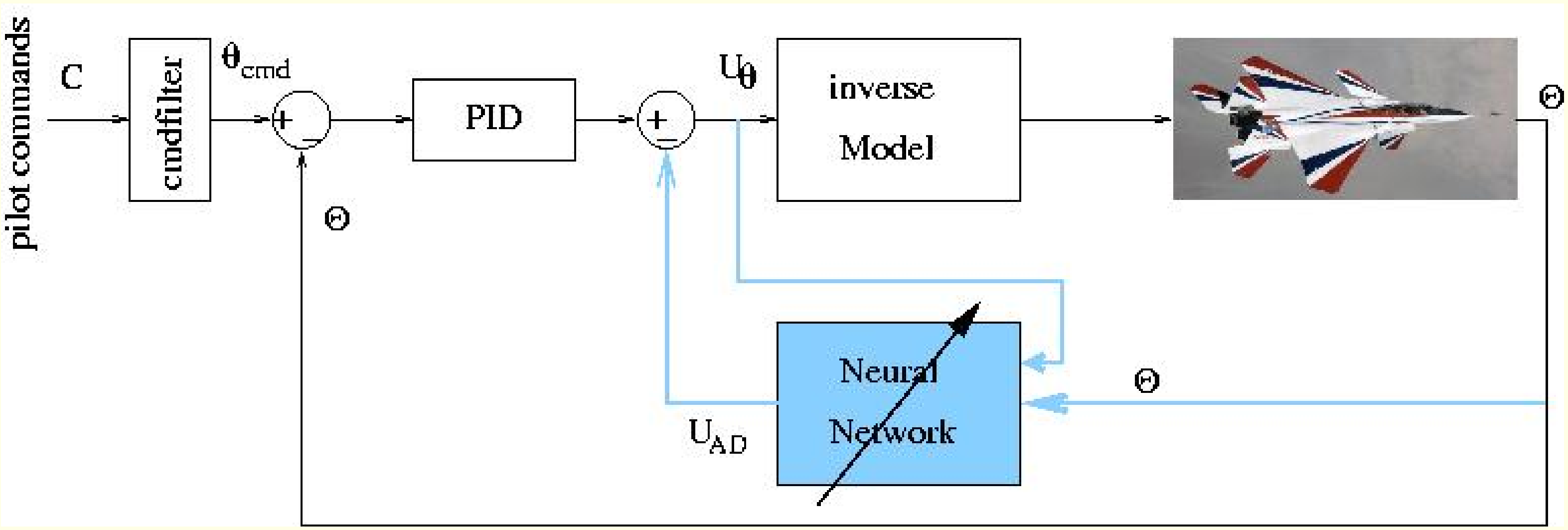
Johann Schumann, RIACS / NASA Ames
Pramod Gupta, QSS / NASA Ames

Adaptive Feedback Control

- Control systems with fixed gain controllers cannot deal with catastrophic changes or degradation in plant
- Adaptive systems (e.g., NN) can react to unexpected situations through learning
- Large potential for adaptive control systems
  – IFCS NN controlled aircraft
  – UAV control, . . .

Adaptive Control

correction of system by *adaptation* of *control law*



- Neural Network produces *correction signal* U_{AD}
- goal: keep deviation as small as possible
- Network is trained (adapted) during operation

Verification & Validation — traditional methods

- Fault avoidance (by design):
  Analysis and Simulation
  - frequency response
  - stability and robustness
  - controllability
  - analysis of covariance
- Fault removal (find and fix problem):
  - testing, testing, and testing
- Fault tolerant (fail-safe) designs:
  - redundancy
  - robustness

While still useful, traditional methods alone are insufficient for verification & validation of adaptive control systems

Verification & Validation — adaptive Control

- Fault avoidance (by design): Analysis and Simulation
  - frequency response
  - stability and robustness
  - controllability
  - analysis of covariance
- Fault removal (find and fix problem):
  - testing, testing, and testing
- Fault tolerant (fail-safe) designs:
  - redundancy
  - robustness

- Applies to base-line case only
  - unanticipated failure?
  - unmodeled failure?
- cannot test all possible configurations in advance
- not possible: fault tolerance under all circumstances

While still useful, traditional methods alone are insufficient for verification & validation of adaptive control systems

Performance Estimation of Neural Network

traditional

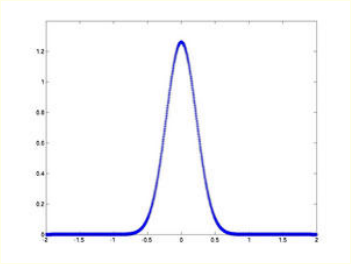
- black box: output is “just the function value”
- no estimate on quality of the NN output

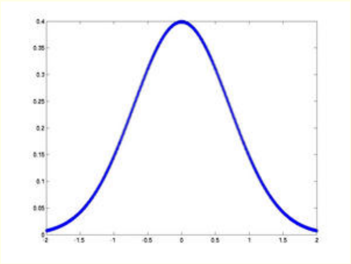
our approach

- black box **+ error bars** (confidence interval on NN outputs)

$$\mathbf{o} \sim N(\mu_o, \sigma^2)$$

- small error bar σ^2 = good quality; large error bar = bad





Our Bayesian Approach

Bayesian analysis provides a proven statistical foundation
on which to judge neural network performance

Basic ideas

- "Engineering Assumption": Data and weights are Gaussian distributed
- Performance measure = standard deviation σ^2 of $p(\mathbf{o}|\mathbf{x}, \mathcal{H})$
- $p(o|\mathbf{x}, \mathcal{H}) = \int p(o|x, \mathbf{w}) p(\mathbf{w}|\mathcal{H})\, d\mathbf{w}$
- where
  - $\mathbf{x}$ network input, $\mathbf{o}$ network output,
  - $\mathcal{H}$ training history,
  - $\mathbf{w}$ network weights

Our Bayesian Approach II

To calculate

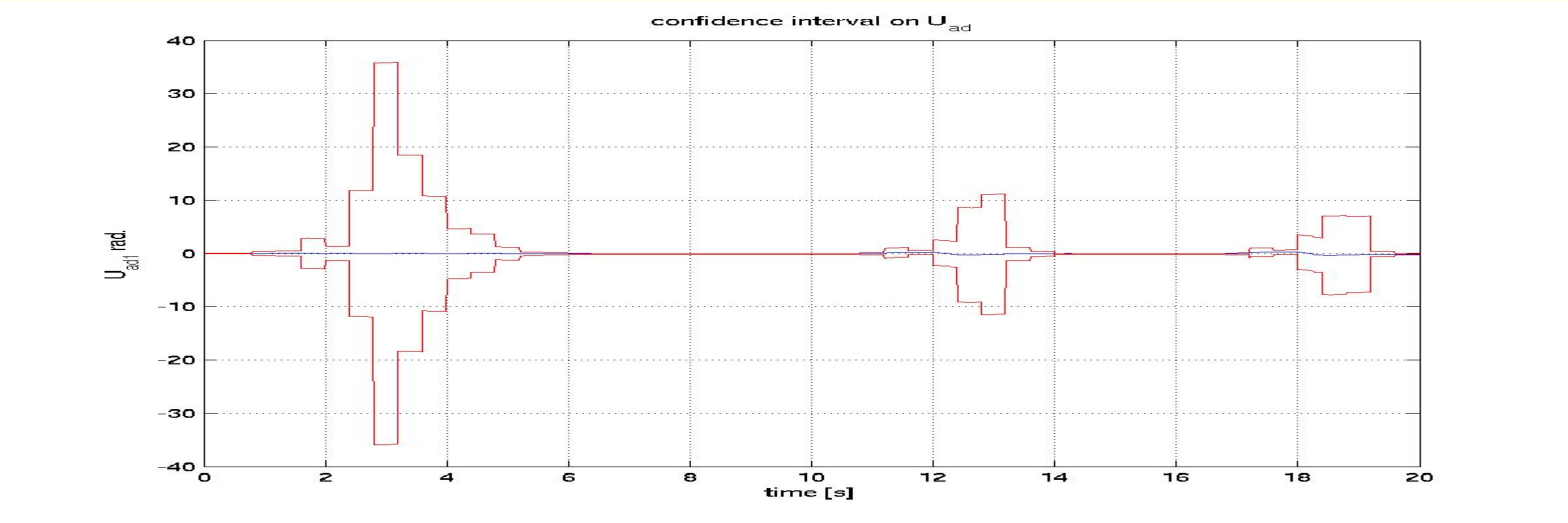
$$p(o|\mathbf{x}, \mathcal{H}) = \int p(o|x, \mathbf{w}) p(\mathbf{w}|\mathcal{H}) \, d\mathbf{w}$$

we obtain the *Posterior* Distribution of the weights after training with training data $\mathcal{H}$, namely $p(\mathbf{w}|\mathcal{H})$ by using Bayes' rule

$$p(\mathbf{w}|\mathcal{H}) = \frac{p(\mathcal{H}|\mathbf{w}) p(\mathbf{w})}{p(\mathcal{H})}$$

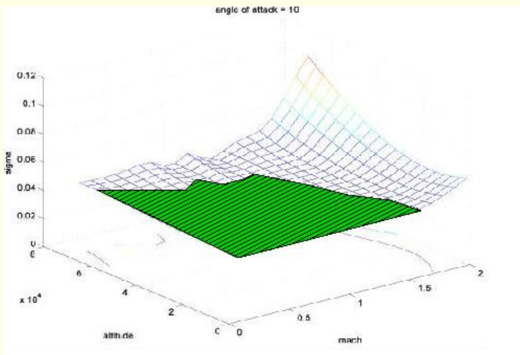
Formula for ${\sigma_t}^2$ depends on current network input, weights, training history, and network architecture.

Performance of Neural Network



- IFCS Gen-II simulator and Confidence Tool (previous work)
- failure (stuck stabilator) at $t = 1.5s$
- blue line: neural network output (U_{AD})
- red line: error bars $\pm\sigma^2$

Envelope Tool



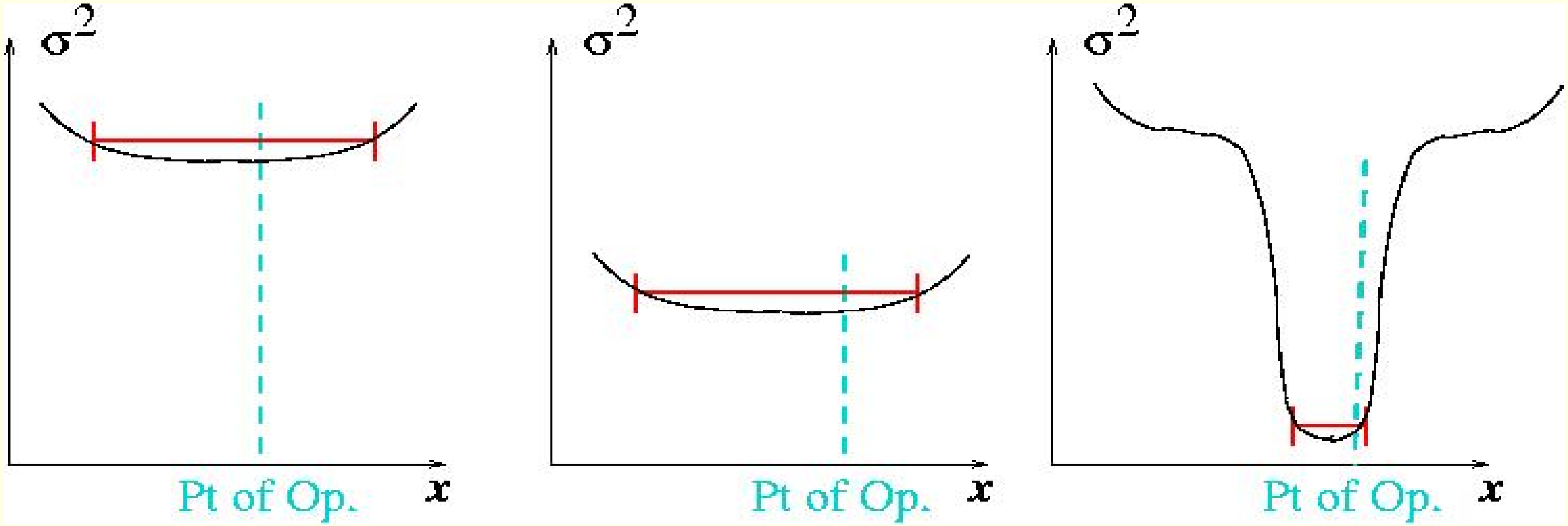
- Lyapunov error bound defines regions of eventual stability
- Regions where confidence is small might cause instability
- Informally: a safe envelope is a region where the confidence level is sufficiently high
- Approach: Bayesian approach combined with sensitivity analysis

Can help answer questions like:

How large is the current safe envelope?

How far is the operational point from the edge?

Interpretation of Some Results

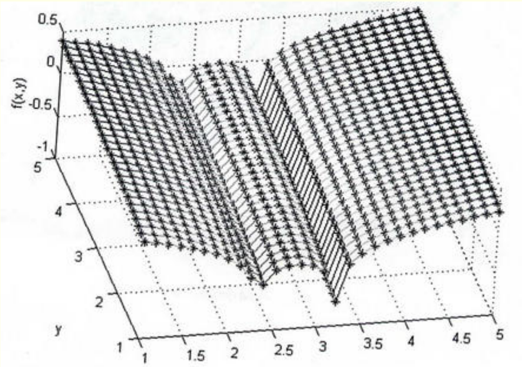


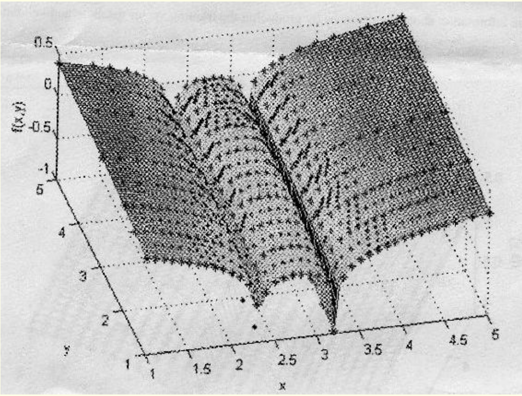
A low network performance, low sensitivity: network needs to adapt

B good performance, low sensitivity: good behavior

C good performance, high sensitivity: network might be overtrained (small changes in operation point lead to drastic performance reduction)

Efficient Calculation of Performance Envelope





- On-going work
- Minimize number of calculations
- Important for dynamic envelope determination
- Based on algorithms from:
  - "Design of Experiments"
  - dynamic gridding
  - computational geometry

Conclusions and Future Work

- Accomplishments
  - Envelope Tool: mathematical background and prototypical Simulink implementation, first experimental results
  - Case Study I: IFCS Gen-II flight control (non-ITAR simulator, Dryden Sim data, test-flight data)
- Future work
  - extend to parameter confidence during system ID
  - extend to other model representation (e.g., interpolation table)
  - relate NN performance measure to system performance
  - Case Study II and III: TBD